CERTIFICATE OF MAILING BY FIRST CLASS MAIL

I hereby certify that this document is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Appeal Brief – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date set forth below.

by Renee D. East
Date of signature and deposit - April 3, 2006

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Everson et al	)	Group Art Unit: 2136
Serial No.: 09/664,893	)	Confirmation No.: 5121
Filed: 9/19/2000	)	Examiner: P. Parthasarathy
For: Authentication, Application-Authorization, and User Profiling Using Dynamic Directory Services	)	Attorney Docket: 1348(16951)

****

APPELLANT'S BRIEF ON APPEAL

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an appeal from the final rejection of the Examiner dated January 26, 2006, rejecting claims 1-12 and 21-36.

REAL PARTY IN INTEREST

The real party in interest in the present appeal is Sprint Communications Company L.P., assignee of the entire right, title, and interest in the present application.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to appellant, the appellant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

STATUS OF CLAIMS

The status of the claims is as follows:

Claims allowed: none.

Claims objected to: none.

Claims rejected: 1-12 and 21-36.

Claims withdrawn: none.

The claims being appealed are: 1-12 and 21-36.

STATUS OF AMENDMENTS

No amendment was filed after final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

The present invention, as represented by claims 1-12 and 21-36, relates to the use of dynamic directory services (DDS) to dynamically store information in a directory server that can be used for authentication, application authorization, and user profiling purposes to eliminate the need for numerous authorization and access control schemes with a single standard directory based set of applications (specification page 1, lines 9-13). Computer users are typically authenticated by supplying user IDs and passwords to

security programs that contain or consult user profiles or databases containing access control information for the users. Since most networks, programs, and applications have their own proprietary access control and security systems, computer users who wish to gain access to more than one network, application, and/or program during a computer session must repeatedly re-enter their user IDs, passwords, etc., each time they attempt to transfer from one network to another or from one application or program to another. This also requires each network, application, and program to have and maintain its own separate access control information for all users (page 1, lines 16-30).

The invention of claims 1-12 and 21-36 uses Dynamic Directory Services (DDS) with a directory access protocol such as the Lightweight Directory Access Protocol (LDAP) to provide authentication and authorization for secured networks, applications, and programs. The DDS stores dynamic information such as session information or user ID information in a directory each time a user logs into the system and then maintains the information in the directory until the user logs out. While the information exists in the directory, it can be queried by any other program, application, or network that uses LDAP or other directory protocol to authenticate or authorize the user for the network or application. The present invention therefore eliminates the need to maintain separate access control systems for each secured network, program, or application (page 2, lines 3-13).

More particularly, claim 1 relates to a method for dynamically tracking a user session in order to authenticate and authorize a computer user to a plurality of separately secured remote applications. Security information is stored for a plurality of computer users in a user profile database (page 4, lines 11-18). When a user launches a first secured computer application on an application server, an authorization server coupled to the user profile database receives login information from the computer user (page 5, line 32 to page 6, line 13). In response to the logging in, a Session ID is created by the authorization server for the computer user (page 6, lines 14-20). At least a portion of the

Session ID is then stored on the user's computer (page 6, lines 21-25). Also in response to the logging in, an object associated with the computer user or the Session ID is created and then the object is dynamically stored in a directory on a directory server coupled with the authorization server and the application server. At least some of the security information relating to the computer user is copied from the user profile database to the object in the directory (page 6, line 26 to page 7, line 2). After comparing the log-in information entered by the computer user to the security information for the computer user, the computer user is allowed access to the first secured computer application if the user is an authenticated or authorized user of the first secured computer application (page 7, lines 3-6). Thereafter, the user launches a second separately-secured computer application on an application server. In response, the second separately-secured computer application reads the Session ID on the user's computer (page 7, lines 7-10). The second separately-secured computer application accesses the object for the computer user on the directory server in response to the Session ID to authenticate or authorize the user for the second separately-secured computer application (page 7, lines 10-15).

The foregoing invention replaces numerous authorization and access control schemes with one standard, directory-based set of applications. The present invention allows all applications, computer programs, and networks that use a directory access protocol such as LDAP to access all user profile and access control information created for a user while the user is logged into the system. This eliminates the need to create and maintain numerous authorization and access control schemes and requires a user to be authorized only once during a computer session.

Claim 7 relates to the system for implementing the above method. Specifically, the system for dynamically tracking a user session in order to authenticate and authorize a computer user to a plurality of separately secured remote applications includes a user profile database 18 (Fig. 1) for storing security information for a plurality of computer users. An authorization server 16 (Fig 1) is coupled with user profile database 18 for

receiving log-in information from a computer user who has launched a first secured computer application, for creating a Session ID for the computer user, for storing at least a portion of the Session ID on the user's computer and for creating an object associated with the computer user or the Session ID. A directory is stored in a directory server 20 (Fig 1) coupled with authorization server 16 for dynamically storing the object created by the authorization server. The authorization server is further operable for copying at least some of the security information relating to the computer user from user profile database 18 to the object in the directory, comparing log-in information entered by the computer user to the security information for the computer user and allowing the computer user access to the launched first secured computer application if the user is an authenticated or authorized user of the computer application. Directory server 20 permits other separately-secured computer applications launched by the computer user to reference the Session ID read by the separately-secured computer applications on the user's computer so that the other separately-secured computer applications may access the object for the computer user on the directory server 20 to authenticate or authorize the user for the other separately-secured computer applications.

Independent claims 27 and 32 relate to a system and method of authenticating and authorizing a user to a plurality of separately-secured computer applications. In particular, the user remotely launches a first secured computer application from a user computer 12 (Page 5, line 32 to page 6, line 3). The user is authenticated and authorized to the first secured computer application by exchanging security information between the user and authorization server 16 (page 6, lines 4-13). At least a portion of the security information is stored in an object within a dynamic directory on a directory server (page 6, lines 29-31). A link to the object is stored on user computer 12 (page 6, lines 21-25). When the user remotely launches a second separately-secured computer application on an application server, the link is retrieved (page 7, lines 8-10 and page 6, lines 23-25). Next, the user is authenticated and authorized to the second separately-secured computer

application by exchanging the stored security information between the directory server and the application server (page 7, lines 10-13).

None of the claims contain either a means plus function or a step plus function element.

GROUNDS OF REJECTION TO BE REVIEWED

1. Whether Claims 1-12 and 21-36 fail to comply with the written description requirement under 35 USC §112, first paragraph.

2. Whether Claims 1-4, 7-10, 21, 24, 27, 29, 30, 32, 34, and 35 are unpatentable under 35 U.S.C. §102(e) as being anticipated by Alegre et al.

3. Whether Claims 5, 6, 11, 12, 31, and 36 are unpatentable under 35 U.S.C. §103(a) as being unpatentable over Alegre et al in view of Hartman et al.

4. Whether Claims 22, 23, 25, 26, 28, and 33 are unpatentable under 35 U.S.C. §103(a) as being unpatentable over Alegre et al in view of Blanco et al.

ARGUMENT

Rejection of Claims 1-12 and 21-36 under 35 USC §112

Claims 1-12 and 21-36

The basis of the rejection is that the "claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the art that the inventor(s), at the time the application was filed, had possession of the claimed invention." As explained below, the claims are fully consistent with the specification and are clearly supported in the manner required by §112.

The rejection argues that the specification does not disclose "a plurality of

separately secured remote applications" or similar references to first and second separately secured applications. In actuality, the plurality of separately secured applications are expressly described throughout the specification. The Description of the Prior Art on page 1 discusses the problem that "computer users who wish to gain access to more than one ... application ... during a computer session to repeatedly re-enter their user IDs ... each time they attempt to transfer from ... one application or program to another." On page 3, it is disclosed that the "application servers 14 are coupled with the user computers 12 via the communications network 22 and are provided for running applications on behalf of the user computers." Thus, the applications that are run by a computer user are remote applications. On pages 5 and 6, it is described how a "user first launches some application or program in a conventional manner" and how the "user next logs into the selected authorization server 16 using account or ID information". Thus, the remote application is a secured application. Page 7 describes that "when the user attempts to access other applications ... while he or she is still logged into the system, these other applications may reference the Session ID ... for authorization purposes related to the new applications." Thus, the specification teaches a plurality of separately secured remote applications.

The rejection also argues that the specification does not indicate how to store a link or retrieve a link. The claims recite storing security information in an object within a dynamic directory on a directory server and then storing a link to the object on the user computer. On page 6, the specification teaches:

> The authorization server 16 then copies or links the Session ID or some derivative thereof to something on the user's computer 12 such as a cookie, shared application memory, or the computer's network address. It is important only that other applications launched by the user from the user computer be able to read or otherwise determine this Session ID by accessing something on the user's computer.

One skilled in the art would reasonably understand that in the act of linking the Session ID to something on the user's computer, a link (which may have the form of a cookie) is created and stored. Once created, other applications may read (retrieve) it. Thus, the claims are fully supported and enabled, and the rejection under 35 USC §112 should be withdrawn.

Rejection of Claims 1-4, 7-10, 21, 24, 27, 29, 30, 32, 34, and 35 under 35 U.S.C. §102(e) as being anticipated by Alegre et al.

Claims 1, 7, 27, and 32

In the method and system of claims 1, 7, 27, and 32, an object associated with the Session ID is stored dynamically in a directory on a directory server coupled with the authorization server. The directory server permits other computer applications launched by the computer user to reference the Session ID on the user's computer. The user is authenticated and authorized to the first secured computer application to be launched by interacting with an authorization server. The user is authenticated and authorized to a second separately-secured computer application by accessing the object for the computer user on the directory server rather than requiring further interaction with the authorization server. The ability of additional applications to authenticate or authorize directly with the directory server achieves important advantages such as reducing network overhead.

Applicant respectfully points out that Alegre et al fails to teach all the claimed limitations, either expressly or implicitly. Alegre et al neither shows nor suggests separately-secured computer applications that are remotely launched by a user. Rather than authenticating and authorizing a user with respect to separately secured applications, Alegre et al creates a session key that is stored at a client browser and is used to access a trusted network. As opposed to authenticating a user to a particular application, Alegre et al requires every message transmitted from the user to the network to be authenticated. Whenever the user accesses the trusted network during the session, the session key must

be transmitted with the access request (col. 3, line 67, to col. 4, line 7). Thus, user authentication is checked for each and every individual remote access request by the user. The session key must be transmitted and checked with every incoming access request from the user, resulting in very high network overhead which is avoided by the present invention. Alegre et al does not have any teaching of authenticating a user to a remote application on an application server, as is required by the present claims.

Alegre et al has no teaching whatsoever of multiple applications that each requires its own separate authorization. Therefore, there is likewise no teaching of using a directory to store an object accessed by more than one application for purposes of authentication.

The rejection relies on Alegre et al at column 8, lines 16-27, to allegedly show multiple applications. The relevant portion states:

> In addition to the check against the access profile received from key server 234, applications requiring extra fine grained access control may use the UID received from key server 234 in combination with a local data base of access rules (not shown) to implement additional access policies.

Besides not providing any description of access rules, Alegre et al does not provide any teaching to show what the "additional access policies" are. There is no teaching or suggestion that the user is authenticated or authorized based on information stored in the key server. Further input (e.g., entry of a password - which is avoided by the present invention) might be expected. Alegre et al only invites speculation regarding this point. On the other hand, it is clear that Alegre et al has no teaching of a second application accessing an object created when an earlier application was launched. Therefore, claims 1-4, 7-10, 21, 24, 27, 29, 30, 32, 34, and 35 are allowable over Alegre et al.

Rejection of Claims 5, 6, 11, 12, 31, and 36 under 35 U.S.C. §103(a)

At pointed out above, Alegre et al fails to either teach or suggest the directory objects or the authentication and authorization of a plurality of remote applications by linking to the directory objects. Claims 5, 6, 11, 12, 31, and 36 further recite that a shopping cart that is stored with a directory object. As a result, different shopping applications can then access the shopping cart. Hartman et al creates a shopping cart for a purchaser. However, the shopping cart is accessed by a single merchant's application, and there is no attempt to construct the shopping cart in a manner that makes it accessible to more than one application. Hartman et al has no directory object of the type claimed in the present application. Neither Hartman et al nor Alegre et al provide any motivation to one skilled in the art to associate a shopping cart with a directory object that can be used by a plurality of applications. Therefore, claims 5, 6, 11, 12, 31, and 36 are allowable over the cited references.

Rejection of Claims 22, 23, 25, 26, 28, and 33 under 35 U.S.C. §103(a)

Claims 22, 23, 25, 26, 28, and 35 recite a dynamic directory service, such as LDAP or X.500. Blanco et al does not use LDAP or X.500 to access objects having the limitations recited in the present claims. Thus, Blanco et al fails to correct for the deficiencies in Alegre et al, and claims 22, 23, 25, and 26 are allowable.

CONCLUSION

The final rejections based on §112, §102, and §103 are improper. The prior art relied upon in the final rejection neither teaches nor suggests the structure or function of the present invention nor does it provide any teaching which can obtain the significant

advantages which are achieved by the present invention. Accordingly, the final rejection dated January 26, 2006, should be reversed.

Respectfully submitted,

Mark L. Mollon
Registration No. 31,123
Attorney for Appellant

Date: April 3, 2006
MacMillan, Sobanski & Todd, LLC
One Maritime Plaza, Fourth Floor
720 Water Street
Toledo, Ohio 43604
Tel: 734-542-0228
Fax: 734-542-9569

CLAIMS APPENDIX

Claims 1-12 and 21-36 now read as follows:

1. A method for dynamically tracking a user session in order to authenticate and authorize a computer user to a plurality of separately secured remote applications, the method comprising the steps of:

a. storing security information for a plurality of computer users in a user profile database;

b. the user launching a first secured computer application on an application server;

c. receiving login information from the computer user at an authorization server coupled with the user profile database;

d. in response to step c, creating a Session ID for the computer user with the authorization server;

e. storing at least a portion of the Session ID on the user's computer;

f. also in response to step c, creating an object associated with the computer user or the Session ID;

g. storing the object dynamically in a directory stored in a directory server coupled with the authorization server and the application server;

h. copying at least some of the security information relating to the computer user from the user profile database to the object in the directory;

i. comparing the log-in information entered by the computer user to the security information for the computer user and allowing the computer user access to the first secured computer application if the user is an authenticated or authorized user of the first secured computer application;

j. the user launching a second separately-secured computer application on an application server;

k. the second separately-secured computer application reading the Session ID on the user's computer; and

l. the second separately-secured computer application accessing the object for the computer user on the directory server in response to the Session ID to authenticate or authorize the user for the second separately-secured computer application.

2. The method as set forth in claim 1, the security information including authentication and authorization information.

3. The method as set forth in claim 2, the authentication and authorization information including at least one of the following: user names, user IDs, passwords, public-key data, certificates, and access control information.

4. The method as set forth in claim 1, the Session ID being based on at least one of the following: a date on which the computer user launched the first secured computer application; a time in which the computer user launched the first secured computer application; a TCP/IP address of the computer user; and a user name of the computer user.

5. The method as set forth in claim 1, further including the steps of creating a shopping cart and storing the shopping cart along with the object in the directory.

6. The method as set forth in claim 5, further including the steps of allowing the user to select items to be purchased and storing information relating to the selected items in the shopping cart.

7. A system for dynamically tracking a user session in order to authenticate

and authorize a computer user to a plurality of separately secured remote applications, the system comprising:

a user profile database for storing security information for a plurality of computer users;

an authorization server coupled with the user profile database for receiving log-in information from a computer user who has launched a first secured computer application, for creating a Session ID for the computer user, for storing at least a portion of the Session ID on the user's computer and for creating an object associated with the computer user or the Session ID; and

a directory stored in a directory server coupled with the authorization server for dynamically storing the object created by the authorization server,

the authorization server being further operable for copying at least some of the security information relating to the computer user from the user profile database to the object in the directory, comparing log-in information entered by the computer user to the security information for the computer user and allowing the computer user access to the launched first secured computer application if the user is an authenticated or authorized user of the computer application,

the directory server permitting other separately-secured computer applications launched by the computer user to reference the Session ID read by the separately-secured computer applications on the user's computer so that the other separately-secured computer applications may access the object for the computer user on the directory server to authenticate or authorize the user for the other separately-secured computer applications.

8. The system as set forth in claim 7, the security information including authentication and authorization information.

9. The system as set forth in claim 8, the authentication and authorization information including at least one of the following: user names, user IDs, passwords, public-key data, certificates, and access control information.

10. The system as set forth in claim 7, the Session ID being based on at least one of the following: a date on which the computer user launched the first secured computer application; a time in which the computer user launched the first secured computer application; a TCP/IP address of the computer user; and a user name of the computer user.

11. The system as set forth in claim 7, the directory server being further operable for creating a shopping cart and storing the shopping cart along with the object in the directory.

12. The system as set forth in claim 11, the directory server being further operable for allowing the user to select items to be purchased and storing information relating to the selected items in the shopping cart.

21. The method as set forth in claim 1, wherein the other computer applications access the object on the directory server using a dynamic directory service.

22. The method as set forth in claim 21, wherein the dynamic directory service comprises the lightweight directory access protocol (LDAP).

23. The method as set forth in claim 21, wherein the dynamic directory service comprises the X.500 access protocol.

24. The system as set forth in claim 7, wherein the other computer applications access the object on the directory server using a dynamic directory service.

25. The system as set forth in claim 24, wherein the dynamic directory service comprises the lightweight directory access protocol (LDAP).

26. The system as set forth in claim 24, wherein the dynamic directory service comprises the X.500 access protocol.

27. A method of authenticating and authorizing a user to a plurality of separately-secured computer applications, the method comprising the steps of:

the user remotely launching a first secured computer application from a user computer;

authenticating and authorizing the user to the first secured computer application by exchanging security information between the user and an authorization server;

storing at least a portion of the security information in an object within a dynamic directory on a directory server;

storing a link to the object on the user computer;

the user remotely launching a second separately-secured computer application on an application server;

retrieving the link; and

authenticating and authorizing the user to the second separately-secured computer application by exchanging the stored security information between the directory server and the application server.

28. The method of claim 27 wherein the exchanging of security information between the directory server and the application server employs a dynamic directory

service.

29. The method of claim 27 wherein the security information includes a Session ID that is stored in the object and in the link.

30. The method of claim 27 further comprising the steps of:

one of the secured computer applications storing application data in the object; and

the other one of the secured computer applications retrieving the application data according to the link.

31. The method of claim 30 wherein the one of the secured computer applications is a shopping application, wherein the stored application data is comprised of shopping cart information; and wherein the other one of the secured computer applications is a check-out application.

32. A system for authenticating and authorizing a user remotely launching secured computer applications from a user computer, the system comprising:

an authorization server for authenticating and authorizing the user to the secured computer applications by exchanging security information between the user and the authorization server when a first secured computer application is launched by the user;

a directory server storing at least a portion of the security information in an object within a dynamic directory, wherein a link to the object is stored on the user computer; and

an application server implementing a second separately-secured computer application for remote launching by the user, wherein the second separately-secured computer application retrieves the link, and wherein the user is authenticated and

authorized to the second separately-secured computer application by exchanging the stored security information between the directory server and the application server.

33. The system of claim 32 wherein the exchanging of security information between the directory server and the application server employs a dynamic directory service.

34. The system of claim 32 wherein the security information includes a Session ID that is stored in the object and in the link.

35. The system of claim 32 wherein one of the secured computer applications stores application data in the object, and wherein the other one of the secured computer applications retrieves the application data according to the link.

36. The system of claim 35 wherein the one of the secured computer applications is a shopping application, wherein the stored application data is comprised of shopping cart information; and wherein the other one of the secured computer applications is a check-out application.

EVIDENCE APPENDIX

No evidence has been submitted under 37 CFR §§1.130, §§1.131, §§1.132, or otherwise.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings and no corresponding decisions rendered.